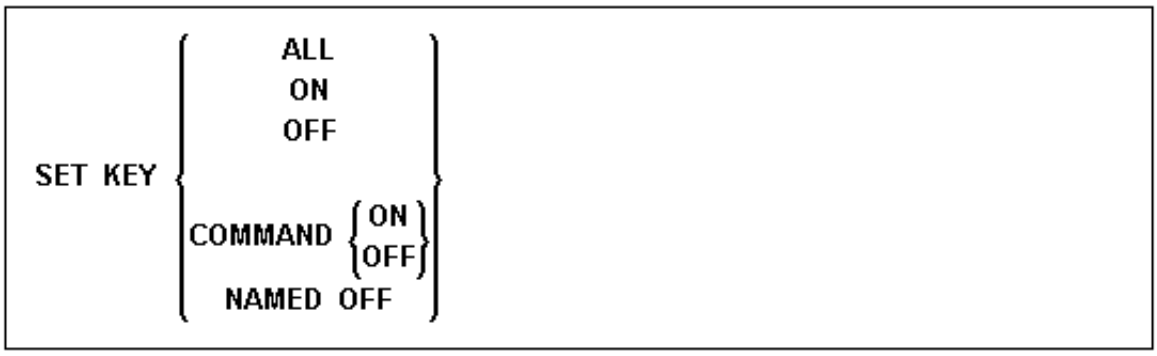
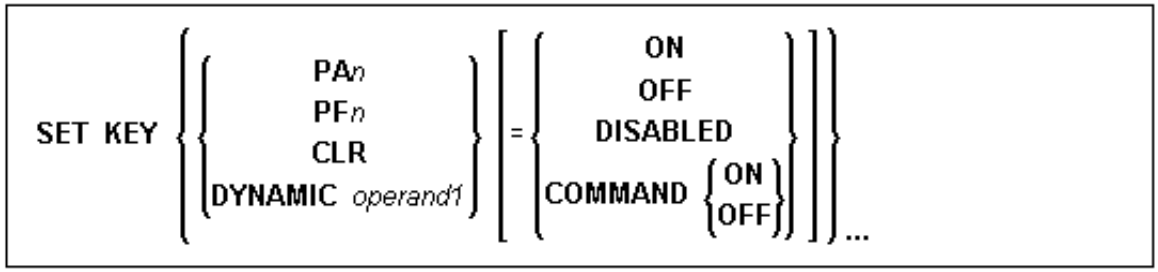
SET KEY

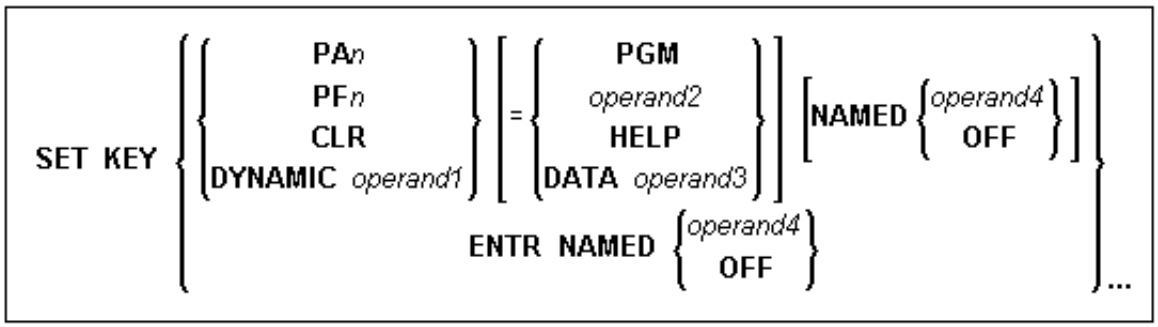
Syntax 1 - Affecting All Keys



Syntax 2 - Affecting Individual Keys



Syntax 3 - Affecting Individual Keys



Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
Operand1		S				A										yes	no
Operand2	C	S				A										yes	no
Operand3	C	S				A										yes	no
Operand4	C	S				A										yes	no

Function

The SET KEY statement is used to assign functions to video terminal PA (program attention) keys, PF (program function) keys, and the CLEAR key.

When a SET KEY statement is executed, Natural receives control of the keys during program execution and uses the values assigned to the keys.

The Natural system variable *PF-KEY identifies which key was pressed last.

Notes:
If a user presses a key to which no function is assigned, either a warning message will be issued prompting the user to press a valid key, or the value "ENTR" will be placed into the Natural system variable *PF-KEY, i.e. Natural will react as if the ENTER key had been pressed (this depends on the Natural profile parameter IKEY as set by the Natural administrator).
On mainframe computers, processing of PA and PF keys is also affected by the Natural profile parameter KEY as set by the Natural administrator.

Making Keys Program-Sensitive

Making a key program-sensitive means that the key will be available for interrogation by the currently active program. If a key is made program-sensitive, pressing the key has the same effect as pressing ENTER. All data that have been entered on the screen are transferred to the program.

Note:
PA keys and the CLEAR key, when made program-sensitive, do not cause any data to be transferred from the screen.

The program-sensitivity remains in effect only for the execution of the current program. See also the section SET KEY Statements on Different Program Levels.

Examples:

SET KEY ALL

This statement causes all keys to be made program-sensitive. All function assignments to any keys are overwritten.

SET KEY PF2
SET KEY PF2=PGM

Each of these statements cause PF2 to be made program-sensitive.

SET KEY OFF

This statement de-activates all key settings. (Under Com-plete on mainframe computers, control of all keys is returned to the TP monitor.)

SET KEY ON

This statement re-activates the functions assigned to all keys that had an assignment and re-activates the program-sensitivity of keys that were made program-sensitive before they were de-activated.

SET KEY PF2=OFF

This statement de-activates PF2. (Under Com-plete on mainframe computers, control of the PF2 key is returned to the TP monitor.)

SET KEY PF2=ON

This statement re-activates the function assigned to PF2 before it was de-activated or made program-sensitive. If no function had been assigned to PF2, it will be made program-sensitive again.

Assigning Commands/Programs

You can assign a command or program name to a key. When the key is pressed, the current program is terminated and the command/program assigned to the key is invoked via the Natural stack. When assigning a command/program, you can also pass parameters to the command/program (see third example below).

You can also assign a terminal command to a key. When the key is pressed, the terminal command assigned to the key is executed.

When operand2 is specified as a constant, it must be enclosed within apostrophes.

Examples:

The command "SAVE" is assigned to PF4:

SET KEY PF4 = ’SAVE’

The value contained in the variable #XYZ is assigned to PF4:

SET KEY PF4 = #XYX

The command "LIST", including the LIST parameters "MAP" and "*", is assigned to PF6:

SET KEY PF6 = 'LIST MAP *'

The terminal command "%%" is assigned to PF2:

SET KEY PF2='%%'

The assignment remains in effect until it is overwritten by another SET KEY statement, until the user logs on to another application, or until the end of the Natural session. See also the section SET KEY Statements on Different Program Levels.

Note:
Before a program invoked via a key is executed, Natural internally issues a BACKOUT TRANSACTION statement.

Assigning Input DATA

You can assign a data string (*operand3*) to a key. When the key is pressed, the data string is placed into the input field in which the cursor is currently positioned, and the data are transferred to the executing program (as if ENTER had been pressed).

When *operand3* is specified as a constant, it must be enclosed within apostrophes.

Example:

SET KEY PF12=DATA 'YES'

For the validity of a DATA assignment, the same applies as for a command assignment, that is, it remains in effect until it is overwritten by another SET KEY statement, until the user logs on to another application, or until the end of the Natural session. See also the section SET KEY Statements on Different Program Levels.

COMMAND OFF/ON

With COMMAND OFF, you can temporarily de-activate any function (command, program, or data) assigned to a key. If the key had been program-sensitive before the function was assigned, COMMAND OFF will make it program-sensitive again.

With a subsequent COMMAND ON, you can re-activate the assigned function again.

Examples:

SET KEY PF4=COMMAND OFF

The function that has been assigned to PF4 is temporarily de-activated; if PF4 had been program-sensitive before the function was assigned, it is now made program-sensitive again.

SET KEY PF4=COMMAND ON

The function assigned to PF4 is re-activated again.

SET KEY COMMAND OFF

All functions assigned to all keys are temporarily de-activated; those keys which had been program-sensitive before functions were assigned to them, are now made program-sensitive again.

SET KEY COMMAND ON

All functions assigned to all keys are re-activated again.

With SET KEY PF*nn*=' ' you can delete the function assigned to a key and at the same time deactivate the program sensitivity of the key.

Assigning HELP

You can assign "HELP" to a key. When the key is pressed, the helproutine assigned to the field in which the cursor is currently positioned will be invoked.

The effect is the same as when entering the help character in the field to invoke help. (The help character - default is "?" - is determined by the Natural profile parameter HI as set by the Natural administrator.)

Example:

SET KEY PF1=HELP

For the validity of a HELP assignment, the same applies as for program-sensitivity, that is, it remains in effect only for the execution of the current program. See also the section SET KEY Statements on Different Program Levels.

DYNAMIC

Instead of specifying a specific key with the SET KEY statement, you can use the DYNAMIC option with a variable (*operand1*), and assign a value (PF*n*, PA*n*, CLR) to this variable in the program. This allows you to specify a function and make it dependent on the program logic which key this function is assigned to.

Example:

```
...
IF ...
   MOVE 'PF4' TO #KEY
ELSE
   MOVE 'PF7' TO #KEY
END-IF
...
SET KEY DYNAMIC #KEY = 'SAVE'
...
```

DISABLED

Graphical user interface (GUI) elements, such as push-buttons, menus, and bitmaps, are implemented as PF keys. With the DISABLED option, you can disable the use the of a GUI element assigned to a PF key. The GUI element will then be displayed in grey and is not available for selection.

To cancel the effect of SET KEY PF*nn*=DISABLED, you use SET KEY PF*nn*=ON.

Example:

SET KEY PF10=DISABLED

Disables the use of the GUI element assigned to PF10.

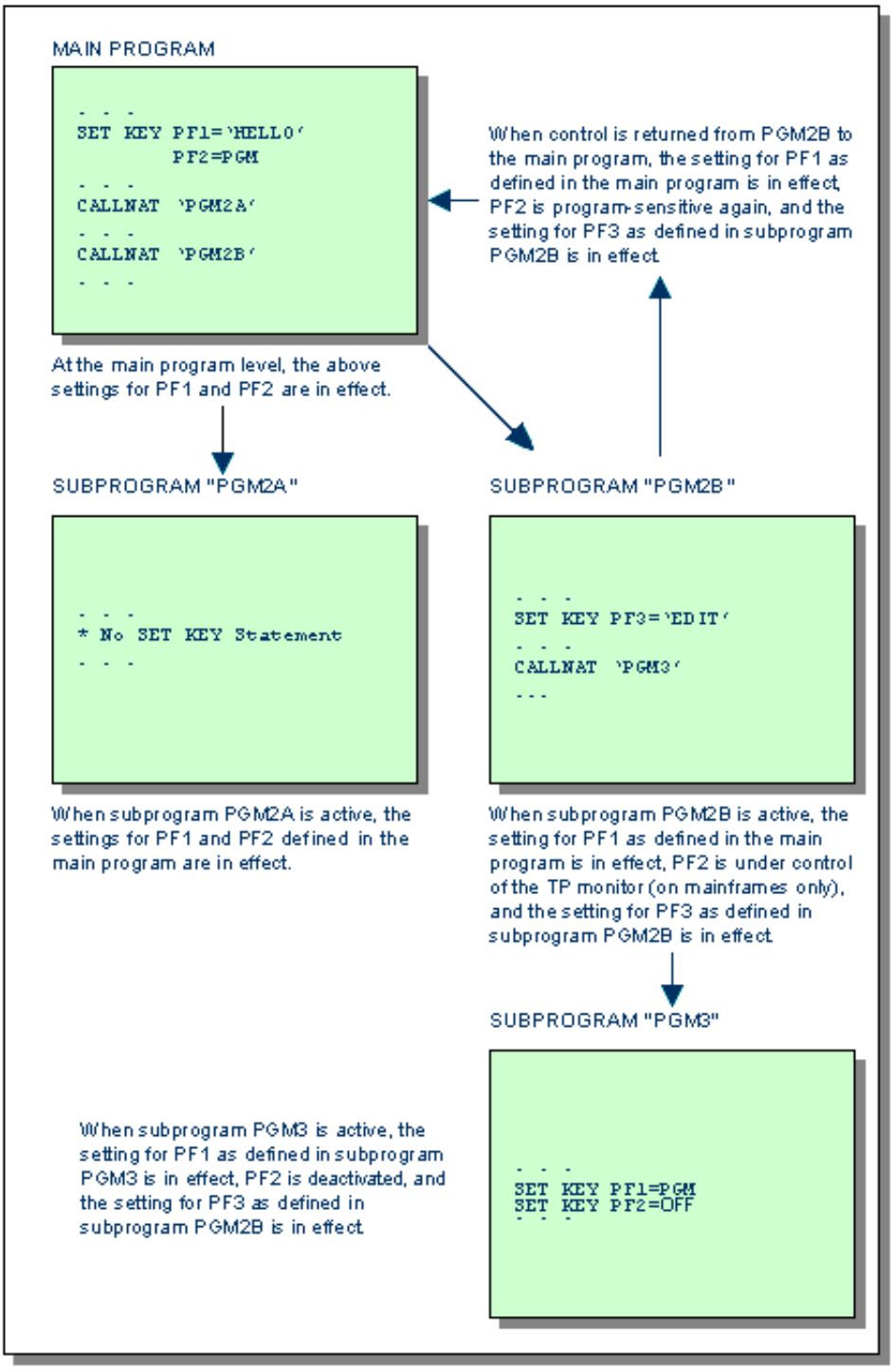
The DISABLED option can only be used within a processing rule.

SET KEY Statements on Different Program Levels

When an application contains SET KEY statements at different levels, the following applies:

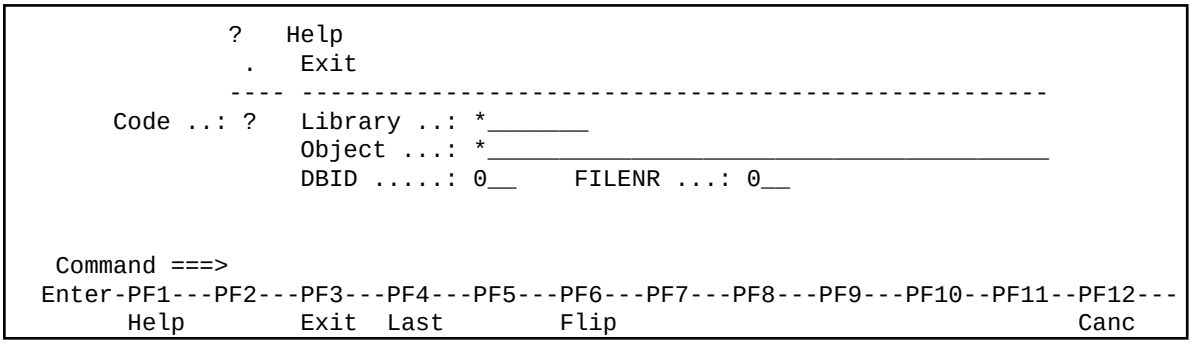
- When keys are made program-sensitive, the program-sensitivity also applies to all lower level (called) programs, unless these programs contain further SET KEY statements. When control is returned to a higher level program, the SET KEY assignments made at the higher level come into effect again.
- For keys which are defined as HELP keys, the same applies as for keys which are program-sensitive.
- When a function (program, command, terminal command, or data string) is assigned to a key, this assignment is valid at all higher and lower levels - regardless of the level at what the assignment is made - until another function is assigned to the key or it is made program-sensitive, or until the user logs on to another application or the Natural session is terminated.

Example of SET KEY Statements on Different Program Levels:



Assigning Names

With the NAMED clause, you can assign a name (*operand4*) to a key. The name will then be displayed in the PF-key lines on the screen; this allows the users to identify the functions assigned to the keys:



The display of the PF-key lines is activated with the session parameter KD (see the Natural Reference documentation). You can control the way in which the PF-key lines are displayed by using the terminal command %Y (see the Natural Reference documentation).

The maximum length of a name to be assigned to a key is 10 characters. In normal tabular PF-key line format (%YN), only the first 5 characters are displayed.

When *operand4* is specified as a constant, it must be enclosed within apostrophes (see examples).

You cannot assign a name to a key without assigning a function to it or making it program-sensitive. To the ENTER key, however, you can only assign a name, but no function.

With NAMED OFF, you delete the name assigned to a program-sensitive key.

Examples:

SET KEY ENTR NAMED ’EXEC’

The name "EXEC" is assigned to the ENTER key.

SET KEY PF3 NAMED ’EXIT’

PF3 is made program-sensitive, and the name "EXIT" is assigned to PF3.

SET KEY PF3 NAMED OFF

PF3 is made program-sensitive, and the name that has been assigned to PF3 is deleted.

SET KEY NAMED OFF

All names that have been assigned to any program-sensitive keys are deleted.

SET KEY PF4=’AP1’ NAMED ’APPL1’

The program "AP1" and the name "APPL1" are assigned to PF4.

When you use normal tabular PF-key line format (%YN), the following applies:

- If you omit the NAMED clause when assigning a command/program to a key, the command/program name will be displayed in the PF-key line; if the command/program name is longer than 5 characters, "CMND" will be displayed.
- If you omit the NAMED clause when assigning input data to a key, "DATA" will be displayed in the PF-key line.

When you use sequential PF-key line format (%YS or %YP), only those keys to which names have been assigned will be displayed in the PF-key line; that is, if you omit the NAMED clause when assigning a command/program/data to a key, the key will not be displayed in the PF-key line.

Example

```
/* EXAMPLE 'SKYEX1': SET KEY
/*****************************************************************
DEFINE DATA LOCAL
1 #PF4 (A56)
1 #FCT (A8)
END-DEFINE
/*****************************************************************
MOVE 'LIST FILES' TO #PF4
/*****************************************************************
SET KEY PF1 PF2
SET KEY PF3 = 'MENU'
        PF4 = #PF4
        PF5 = 'LIST FILE EMPLOYEES'
/*****************************************************************
INPUT 10X 'THE FOLLOWING FUNCTION KEYS ARE AVAILABLE:' //
      10X 'PF1: EMPLOYEES UPDATE PROGRAM' /
      10X 'PF2: EMPLOYEES READ   PROGRAM' /
      10X 'PF3: RETURN TO MENU          ' /
      10X 'PF4: LIST FILES              ' /
      10X 'PF5: LIST FILE EMPLOYEES     ' ///
      10X '     OR YOU MAY ENTER A PROGRAM TO BE EXECUTED:' #FCT
/*****************************************************************
IF #FCT NE ' '
 FETCH #FCT
END-IF
IF *PF-KEY = 'PF1'
 FETCH 'UPDPERS'
END-IF
IF *PF-KEY = 'PF2'
 FETCH 'READPERS'
END-IF
/*****************************************************************
END
```

```
         THE FOLLOWING FUNCTION KEYS ARE AVAILABLE:

           PF1: EMPLOYEES UPDATE PROGRAM
           PF2: EMPLOYEES READ   PROGRAM
           PF3: RETURN TO MENU
           PF4: LIST FILES
           PF5: LIST FILE EMPLOYEES


                OR YOU MAY ENTER A PROGRAM TO BE EXECUTED:
```